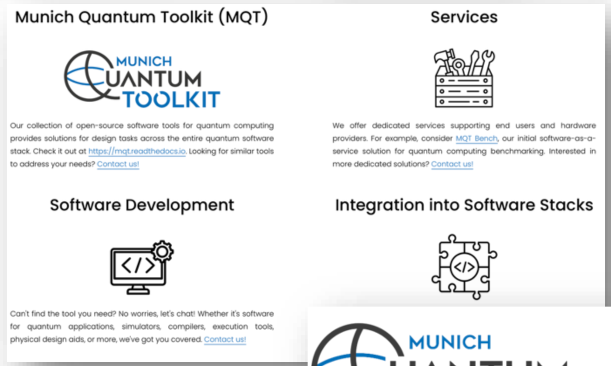
QRUCH Workshop @ ISC25 – June 2025

Connecting Quantum Hardware and Software

The Quantum Device Management Interface (QDMI)

Dr. Lukas Burgholzer
lukas.burgholzer@tum.de

Who am I?


Munich Quantum Toolkit (MQT)

Our collection of open-source software tools for quantum computing provides solutions for design tasks across the entire quantum software stack. Check it out at https://mqt.readthedocs.io. Looking for similar tools to address your needs? Contact us!

Services

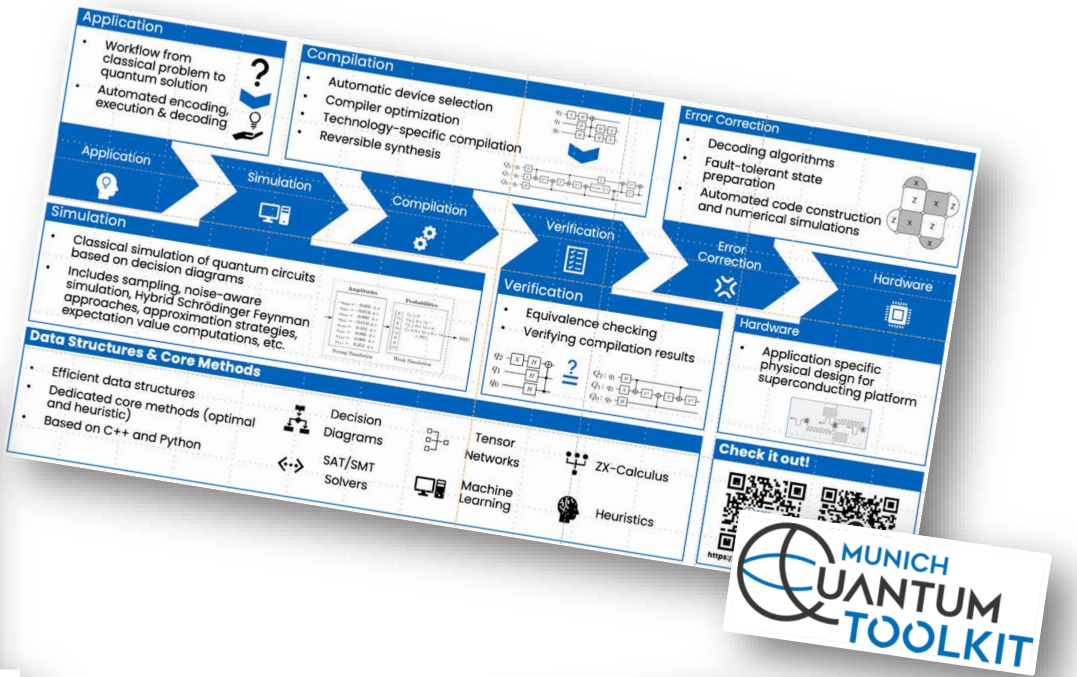
We offer dedicated services supporting end users and hardware providers. For example, consider MQT Bench, our initial software-as-a-service solution for quantum computing benchmarking. Interested in more dedicated solutions? Contact us!

Software Development

Can't find the tool you need? No worries, let's chat! Whether it's software for quantum applications, simulators, compilers, execution tools, physical design aids, or more, we've got you covered. Contact us!

Integration into Software Stacks

MUNICH QUANTUM SOFTWARE COMPANY


Application
- Workflow from classical problem to quantum solution
- Automated encoding, execution & decoding

Compilation
- Automatic device selection
- Compiler optimization
- Technology-specific compilation
- Reversible synthesis

Error Correction
- Decoding algorithms
- Fault-tolerant state preparation
- Automated code construction and numerical simulations

Simulation
- Classical simulation of quantum circuits based on decision diagrams
- Includes sampling, noise-aware simulation, Hybrid Schrödinger Feynman approaches, approximation strategies, expectation value computations, etc.

Verification
- Equivalence checking
- Verifying compilation results

Hardware
- Application specific physical design for superconducting platform

Data Structures & Core Methods
- Efficient data structures
- Dedicated core methods (optimal and heuristic)
- Based on C++ and Python

Check it out!

MUNICH QUANTUM TOOLKIT

Munich Quantum Valley

No matter how cold it gets, there is always this person

MQSS Munich Quantum Software Stack

lrz MQSS Munich Quantum Software Stack Munich Quantum Valley

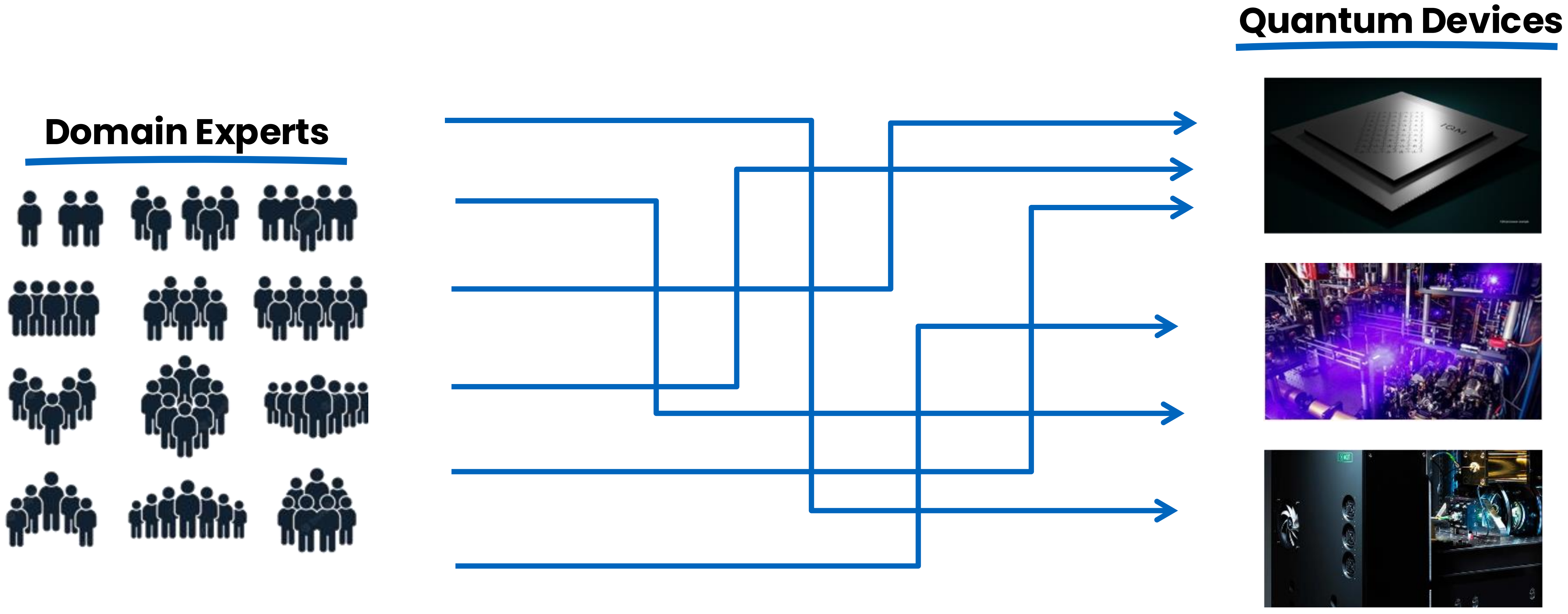
QRUCH Workshop @ ISC25 – June 2025

Connecting Quantum Hardware and Software

The Quantum Device Management Interface (QDMI)

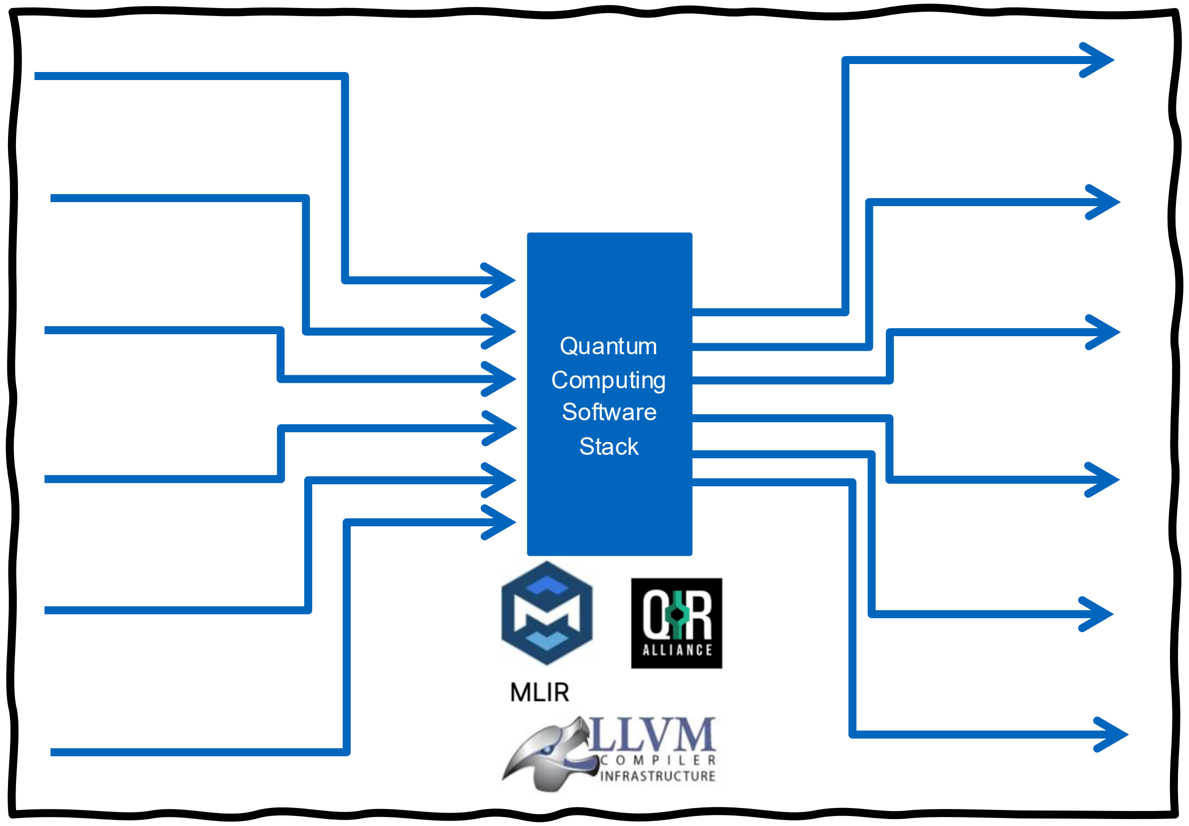
Dr. Lukas Burgholzer
lukas.burgholzer@tum.de

Quantum Computing – The Big Picture

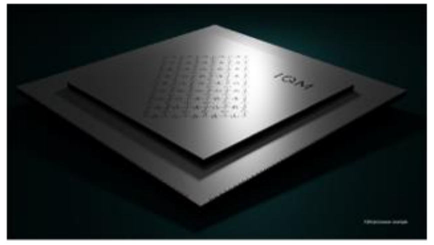


Quantum Computing – The Big Picture

Domain Experts

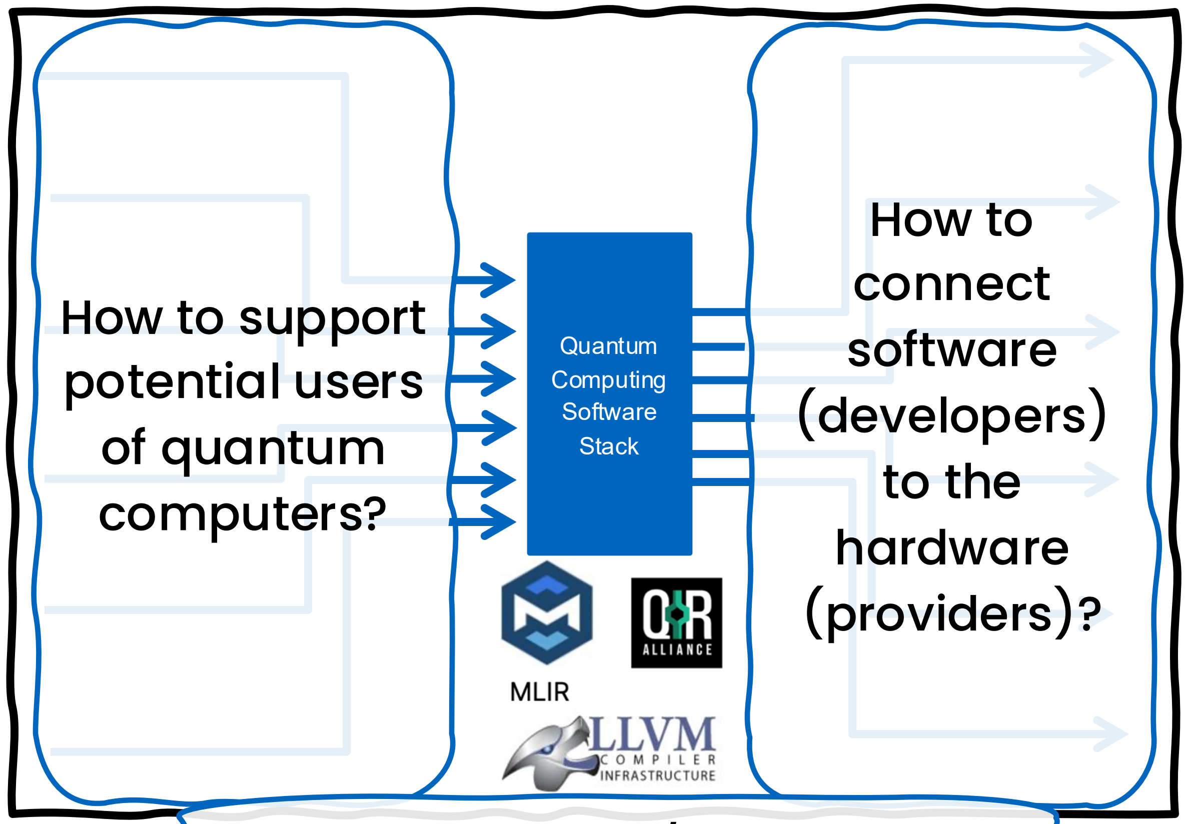

Quantum Computing Software Stack
MLIR
QIR ALLIANCE
LLVM COMPILER INFRASTRUCTURE

Quantum Devices



Quantum Computing – The Big Picture

Domain Experts

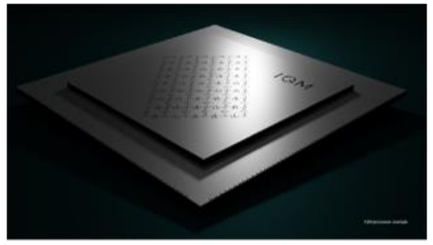

Quantum Computing Software Stack
MLIR
QIR ALLIANCE
LLVM COMPILER INFRASTRUCTURE

How to support potential users of quantum computers?

How to connect software (developers) to the hardware (providers)?

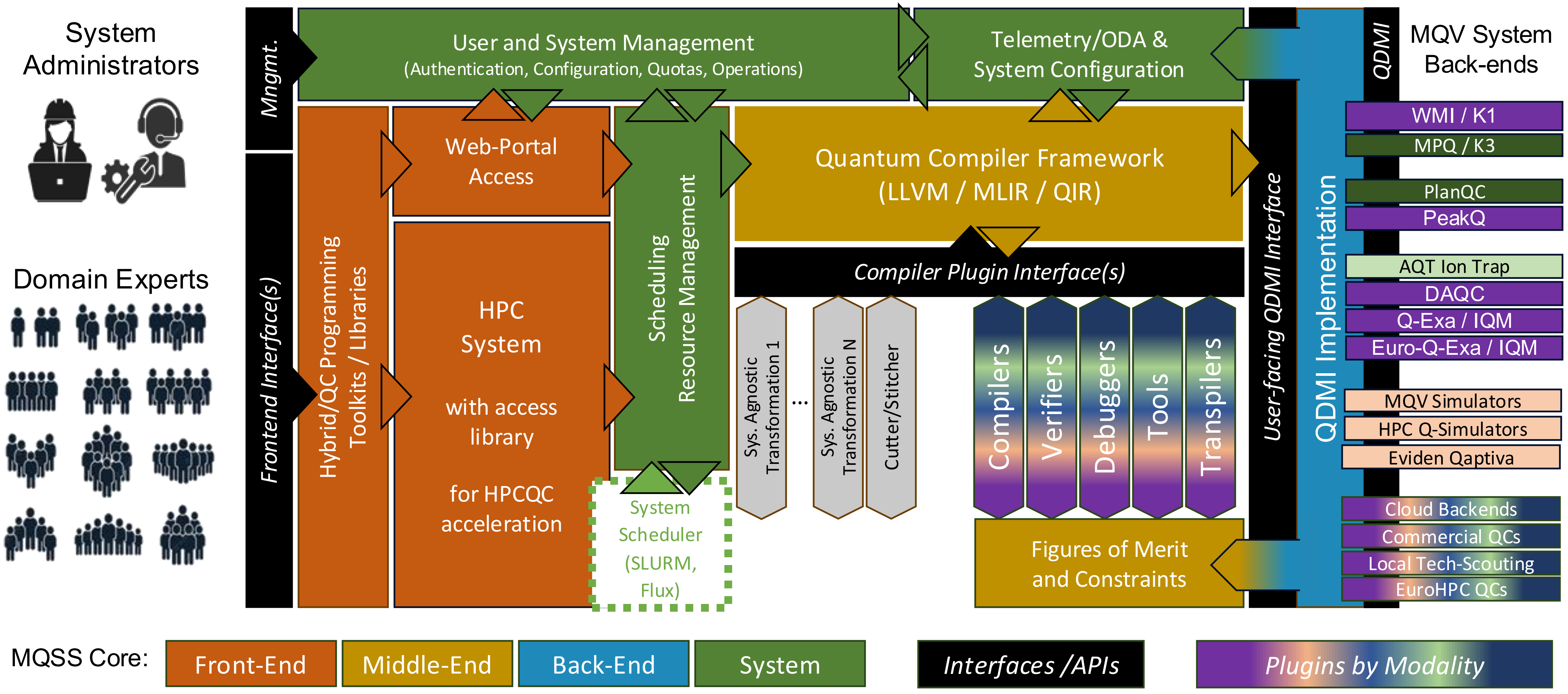
How to connect to/integrate with existing compute & HPC to enable quantum acceleration?

Quantum Devices



MQSS Munich Quantum Software Stack

munich-quantum-valley.de/
research/research-areas/mqss



The Munich Quantum Toolkit (MQT)

All tools are available as open-source repositories on GitHub under the MIT license

MQT ProblemSolver — Application
A Tool for Solving Problems Using Quantum Computing
github.com/cda-tum/mqt-problemsolver
PyPI

MQT Bench — Application
A Quantum Circuit Benchmark Suite
www.cda.cit.tum.de/mqtbench
github.com/munich-quantum-toolkit/bench
PyPI

MQT Quantum Auto Optimizer — Application
An Automatic Framework for Solving Optimization Problems
github.com/cda-tum/mqt-qao
PyPI

MQT QUBOMaker — Application
A Framework for the Automatic Generation of QUBO Formulations
github.com/cda-tum/mqt-qubomaker
PyPI

MQT DDSIM — Simulation
A Tool for Classical Quantum Circuit Simulation based on Decision Diagrams
github.com/cda-tum/mqt-ddsim
PyPI

MQT Predictor — Compilation
A Tool for Determining Good Quantum Circuit Compilation Options
github.com/cda-tum/mqt-predictor
PyPI

MQT Ion Shuttler — Compilation
A Tool for Generating Shuttling Schedules for QCCD Architectures
github.com/cda-tum/ion-shuttler

MQT Qudits — Compilation
A Tool for Compiling High-Dimensional Quantum Systems
github.com/cda-tum/mqt-qudits
PyPI

MQT SyReC — Compilation
A Tool for the Synthesis of Reversible Circuits/Quantum Computing Oracles
github.com/munich-quantum-toolkit/syrec
PyPI

MQT QMAP — Compilation
A Tool for Quantum Circuit Mapping And Clifford Circuit Optimization/Synthesis
github.com/cda-tum/mqt-qmap
PyPI

MQT QCEC — Verification
A Tool for Quantum Circuit Equivalence Checking
github.com/munich-quantum-toolkit/qcec
PyPI

MQT DASQA — Hardware
A Tool for Designing Alternative Superconducting Quantum Architectures
github.com/cda-tum/mqt-dasqa

MQT DDVis — Data Structures
A Web-Application visualizing Decision Diagrams for Quantum Computing
www.cda.cit.tum.de/app/ddvis

MQT Core — Data Structures
The Backbone of the MQT Intermediate Representation (IR) Decision Diagram and ZX Package
github.com/munich-quantum-toolkit/core
PyPI

MQT QuSAT — Core Methods
A Tool for Encoding Quantum Computing using Satisfiability Testing (SAT) Techniques
$F \wedge (x_1 \wedge \neg x_2)$
$F \wedge (x_3 \wedge x$
github.com/munich-quantum-toolkit/qusat
PyPI

MQT QECC — QECC
A Tool for Quantum Error Correcting Codes
github.com/cda-tum/mqt-qecc
PyPI

https://mqt.readthedocs.io

Over 1k ⭐ on GitHub

MUNICH QUANTUM TOOLKIT @ GitHub

Over 2 Million Downloads on PyPI

MUNICH QUANTUM TOOLKIT @ python Package Index

lrz MQSS Munich Quantum Software Stack Munich Quantum Valley TUM

MQSS Components Catalog

munich-quantum-valley.de/research/research-areas/mqss

Front-End

QPI: Hybrid Programming from C/C++
- LRZ/LS & TUM/MS: Ercüment Kaya

FPQA Compiler for Max3SAT problems
- TUM/PB: Oğuzcan Kırmemiş

qTPU: Large circuits as tensor networks
- TUM/PB: Nathaniel Tornow

ISV Job execution for Spin Hamiltonians
- LRZ/LS: Burak Mete and Tobias Bauer

MQT QECC: EC quantum circuit preparation
- TUM/RW: Lucas Berent

Parallel circuit extraction from ZX Diagrams
- LMU/DK: Karl Führlinger

GA4QCD: Application-specific synthesis
- LMU/CLP: Leo Sünkel

qcd-gym: Circuit builder/optimizer using RL
- LMU/CLP: Philipp Altmann

Middle-End

MQT Predictor: Predict suitable back-ends
- TUM/RW: Nils Quetschlich

MILQ: Assigning circuits backends
- TUM/CM: Philipp Seitz and Manuel Geiger

AI-based compiler path selection
- LRZ/LS & TUM/MS: Aleksandra Świerkowska

MQT QMAP: Topology mapping of circuits
- TUM/RW: Lukas Burgholzer

MQT QCEC: Tool for equivalence checking
- TUM/RW: Lukas Burgholzer

MQT Qudits: Compilation for multistate Qbits
- TUM/RW: Kevin Mato

Quantum constant propagation
- TUM/HS: Yanbin Chen

Mid-Circuit measurement reduction
- TUM/HS: Innocenzo Fulginiti

Back-End

Hardware backend development with partners
- LRZ/LS: Jorge Echavarria

FoMaCs via Sys-Sage tool library
- TUM/MS: Stepan Vanecek

Unified Quantum Platform (UQP)
- TUM/MS: Amr Elsharkawy

Quantum Control Processor (QCP) and ISA
- TUM/MS: Xiaorang Guo

Simulator: MQT DDSIM
- TUM/RW: Lukas Burgholzer

Simulator: Tensor networks
- TUM/CM: M. Geiher and Q. Huang

Simulator: Parallel Clifford+T
- LMU/DK: Florian Kroetz

Simulator: Back-ends for HPC simulators
- LRZ/LS: Marco De Pascale

System

Munich Quantum Portal (MQP) and plugins
- LRZ/LS: Marco De Pascale

Resource prediction and circuit scheduler
- LRZ/LS: Minh Chung

IoT Environment / ODA / Digital Twins
- LRZ/LS & TUM/MS: H. Ahmed and Y. Gambo

HPC scheduling
- LRZ/LS & TUM/MS: Nufail Farooqi

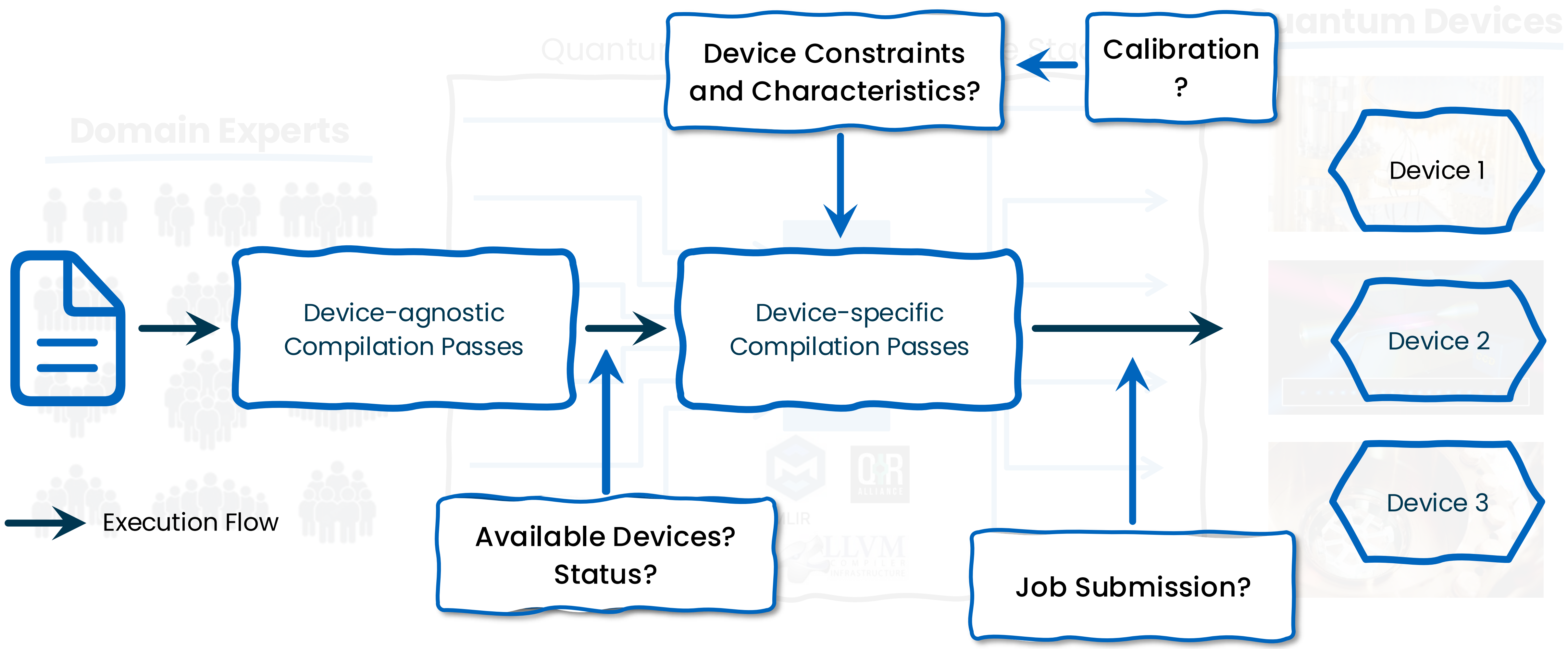
Operations, Configuration, Calibration
- LRZ/LS: Matt Tovey and Xiaolang Deng

QRUCH Workshop @ ISC25 – June 2025

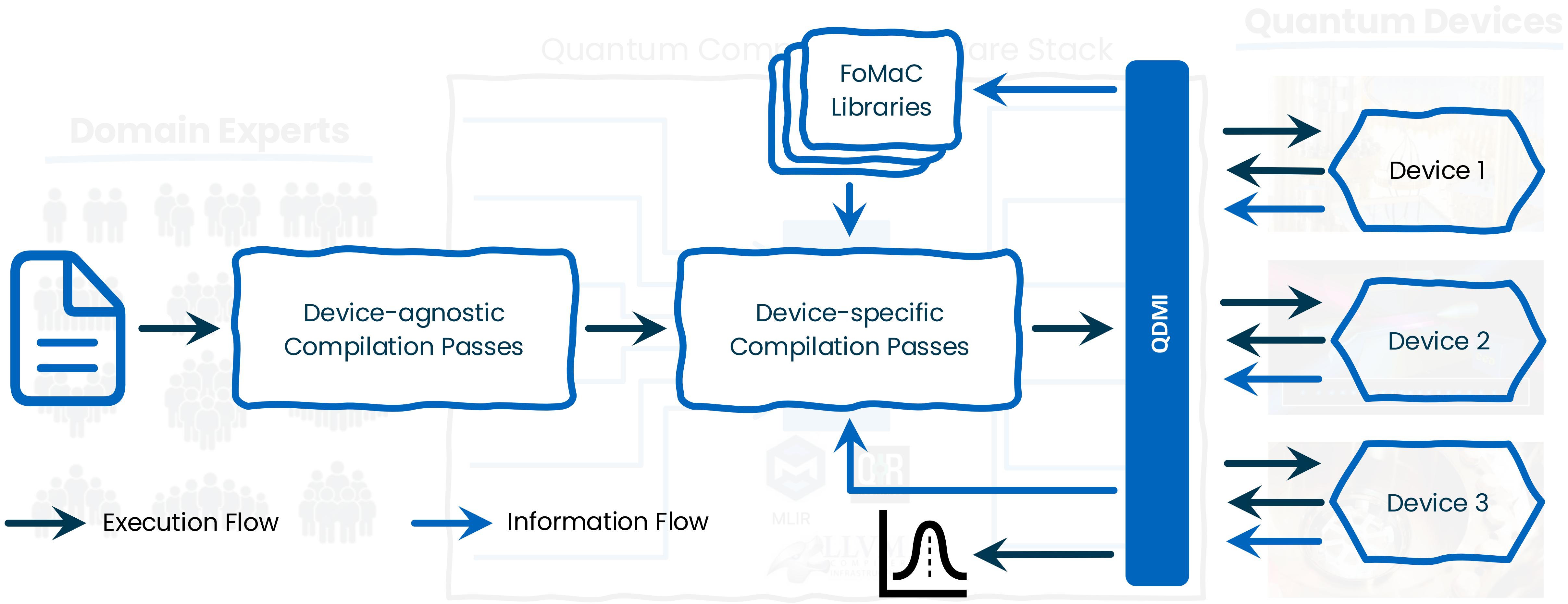
The Quantum Device Management Interface (QDMI)

Dr. Lukas Burgholzer
lukas.burgholzer@tum.de

QDMI Quantum Device Management Interface

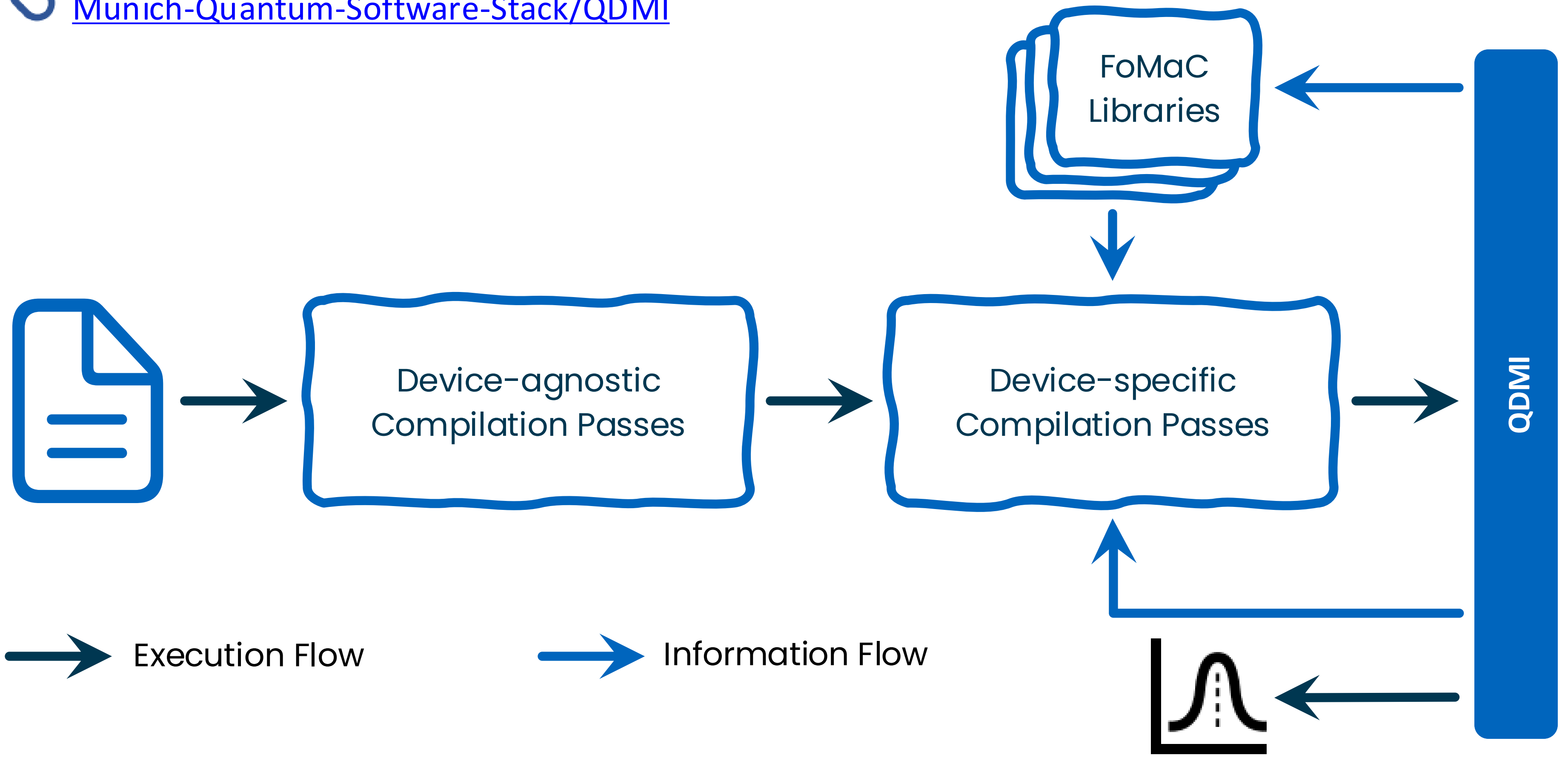

Device Constraints and Characteristics?
Calibration ?
Device-agnostic Compilation Passes
Device-specific Compilation Passes
Device 1
Device 2
Device 3
Execution Flow
Available Devices? Status?
Job Submission?

QDMI Quantum Device Management Interface



QDMI Quantum Device Management Interface

[Github.com/ Munich-Quantum-Software-Stack/QDMI](Github.com/Munich-Quantum-Software-Stack/QDMI)



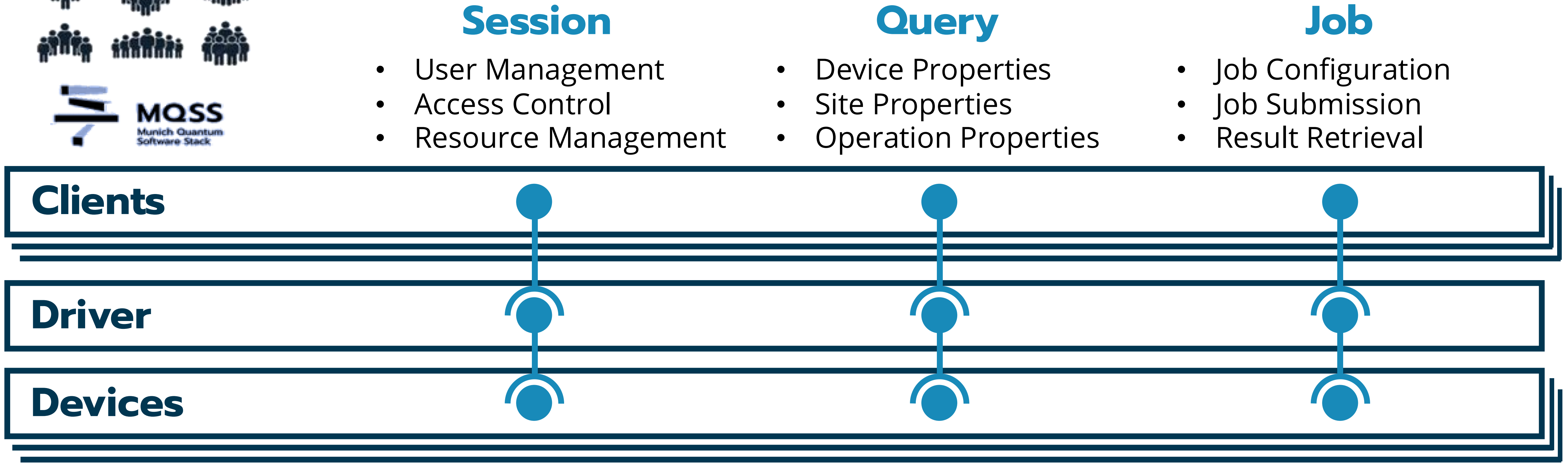
FoMaC Libraries
Device-agnostic Compilation Passes
Device-specific Compilation Passes
QDMI
Execution Flow
Information Flow

Available Devices

- WMI / K1
- MPQ / K3
- PlanQC
- PeakQ
- AQT Ion Trap
- DAQC
- Q-Exa / IQM
- Euro-Q-Exa / IQM
- MQV Simulators
- HPC Q-Simulators
- Eviden Qaptiva
- Cloud Backends
- Commercial QCs
- Local Tech-Scouting
- EuroHPC QCs

open-source, openly-developed, multi-modality, HPC-compatible

QDMI Quantum Device Management Interface



QDMI Quantum Device Management Interface

ession

Management

s Control

rce

Query

- Device Properties
- Site Properties
- erties

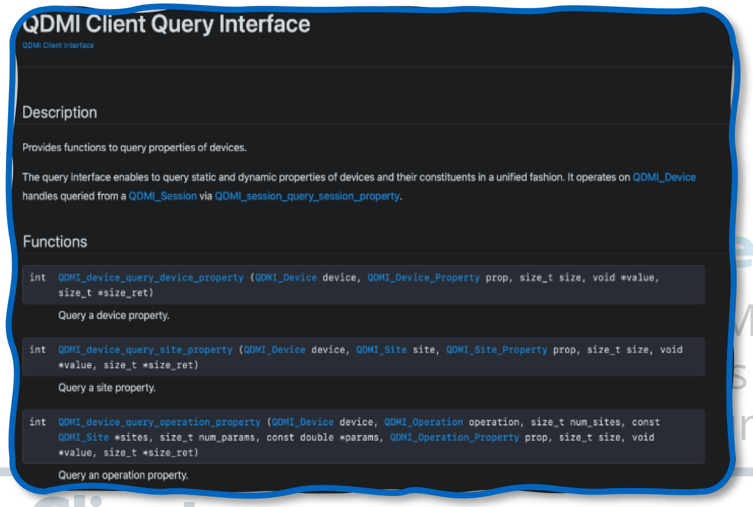
Job

- Job Configuration
- Job Submission
- Result Retrieval

Clients

Driver

Devices



QDMI Client Query Interface

QDMI Client Interface

Description

Provides functions to query properties of devices.

The query interface enables to query static and dynamic properties of devices and their constituents in a unified fashion. It operates on QDMI_Device handles queried from a QDMI_Session via QDMI_session_query_session_property.

Functions

```
int  QDMI_device_query_device_property (QDMI_Device device, QDMI_Device_Property prop, size_t size, void *value,
     size_t *size_ret)
```

Query a device property.

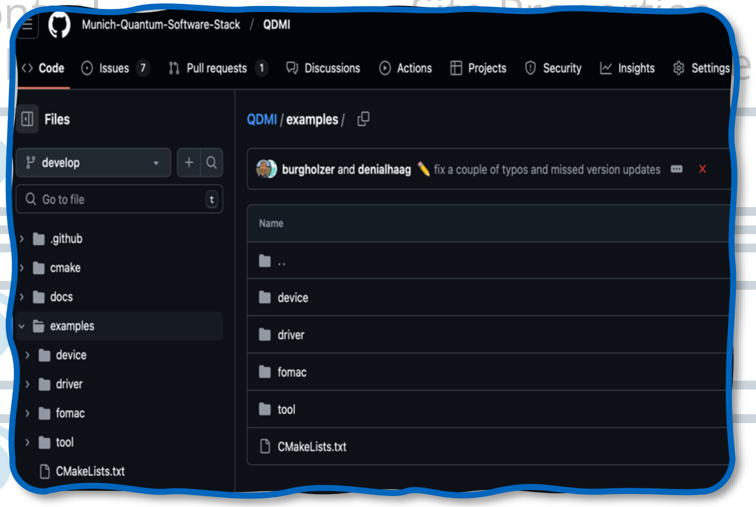
```
int  QDMI_device_query_site_property (QDMI_Device device, QDMI_Site site, QDMI_Site_Property prop, size_t size, void
     *value, size_t *size_ret)
```

Query a site property.

```
int  QDMI_device_query_operation_property (QDMI_Device device, QDMI_Operation operation, size_t num_sites, const
     QDMI_Site *sites, size_t num_params, const double *params, QDMI_Operation_Property prop, size_t size, void
     *value, size_t *size_ret)
```

Query an operation property.

Full API Documentation



Munich-Quantum-Software-Stack / QDMI

Code | Issues 7 | Pull requests 1 | Discussions | Actions | Projects | Security | Insights | Settings

Files

develop

Go to file

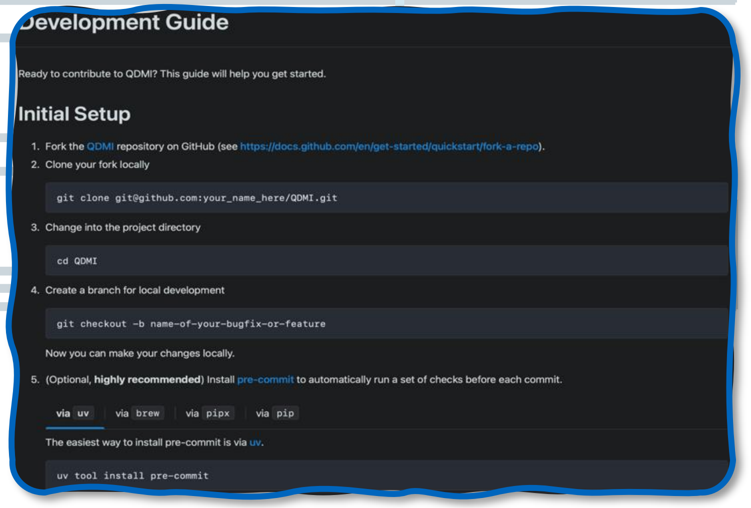
```
> .github
> cmake
> docs
v examples
  > device
  > driver
  > fomac
  > tool
  CMakeLists.txt
```

QDMI / examples /

burgholzer and denialhaag ✏️ fix a couple of typos and missed version updates

Name
..
device
driver
fomac
tool
CMakeLists.txt

Reference Implementations



Development Guide

Ready to contribute to QDMI? This guide will help you get started.

Initial Setup

1. Fork the QDMI repository on GitHub (see https://docs.github.com/en/get-started/quickstart/fork-a-repo).
2. Clone your fork locally

```
git clone git@github.com:your_name_here/QDMI.git
```

3. Change into the project directory

```
cd QDMI
```

4. Create a branch for local development

```
git checkout -b name-of-your-bugfix-or-feature
```

Now you can make your changes locally.

5. (Optional, **highly recommended**) Install pre-commit to automatically run a set of checks before each commit.

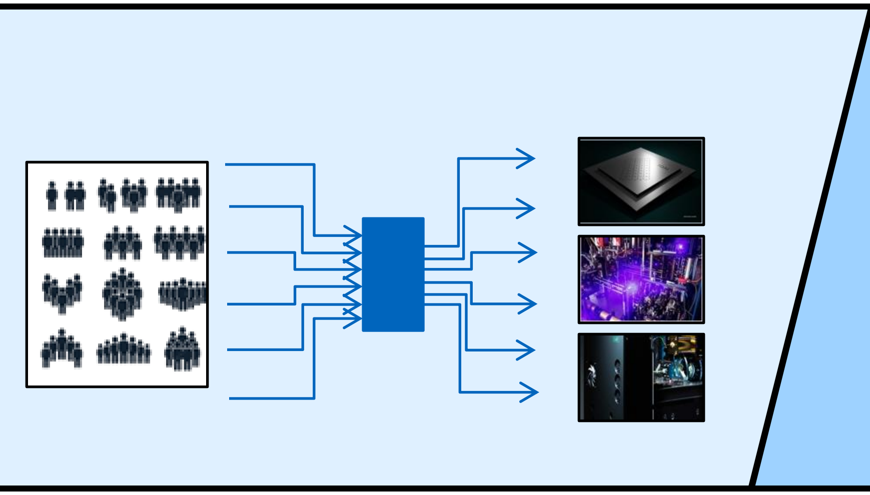
via `uv` | via `brew` | via `pipx` | via `pip`

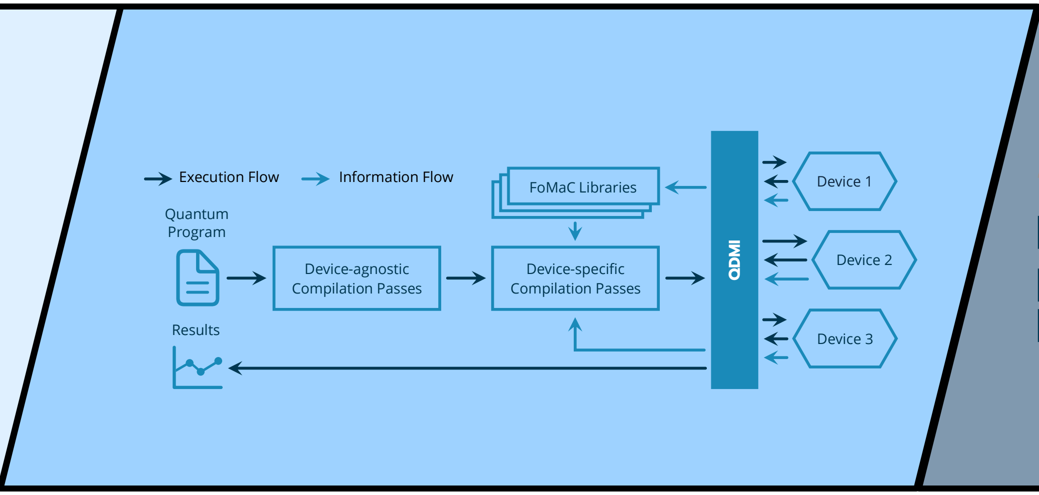
The easiest way to install pre-commit is via uv.

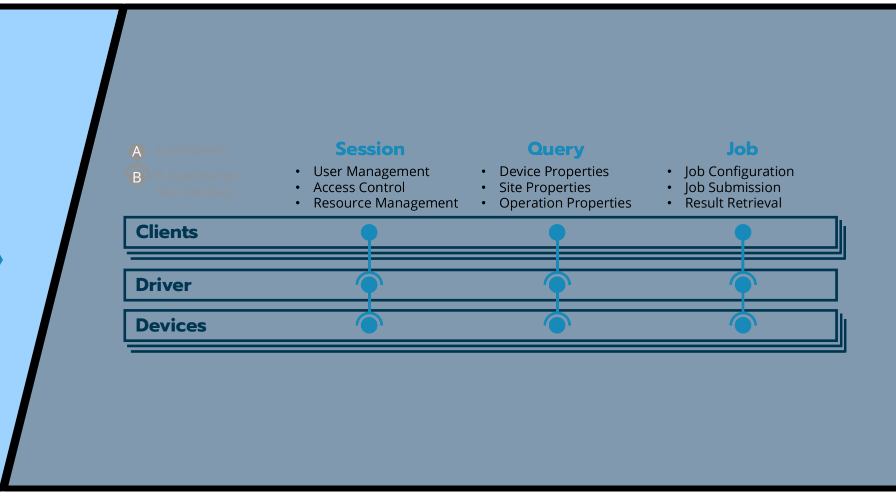
```
uv tool install pre-commit
```

Development Guide

Conclusions







open-source, openly-developed, multi-modality, HPC-compatible

munich-quantum-valley.de/research/research-areas/mqss

github.com/Munich-Quantum-Software-Stack/QDMI

World of Quantum
JUNE 24–27, 2025 I MESSE MÜNCHEN

MUNICH QUANTUM SOFTWARE FORUM
20-21 of OCTOBER 2025

SC25
St. Louis, MO | hpc ignites.

IEEE QUANTUM WEEK

lrz · MQSS Munich Quantum Software Stack · Munich Quantum Valley · TUM